III YEAR V SEMESTER

DATABASE MANAGEMENNT SYSTEM LAB

1. Railway Reservation

AIM: To drwa ER diagrams for train services in railway station .

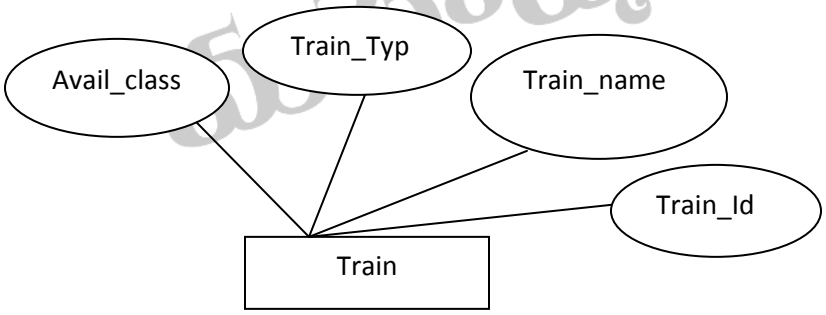
The following are the entities:

1. Train
2. Train status
3. Route
4. Station
5. Passenger
6. User

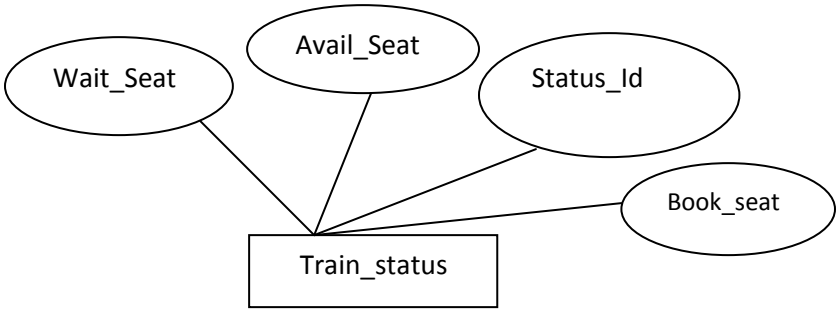
Procedure

The attributes in the entities

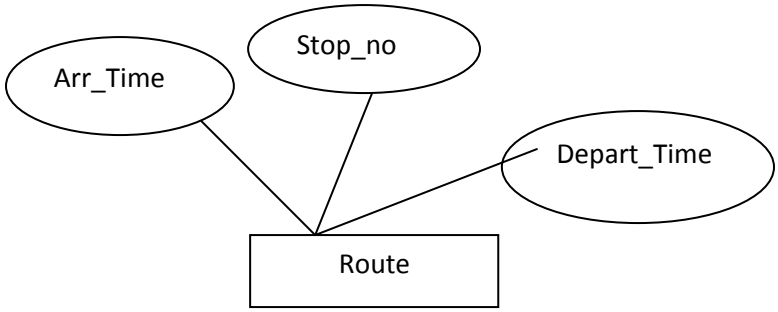
Train (Entitiy)



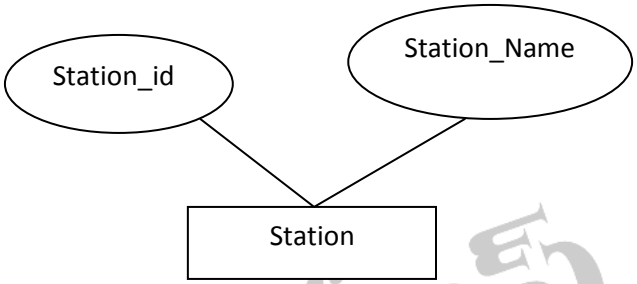
Train status (Entity)



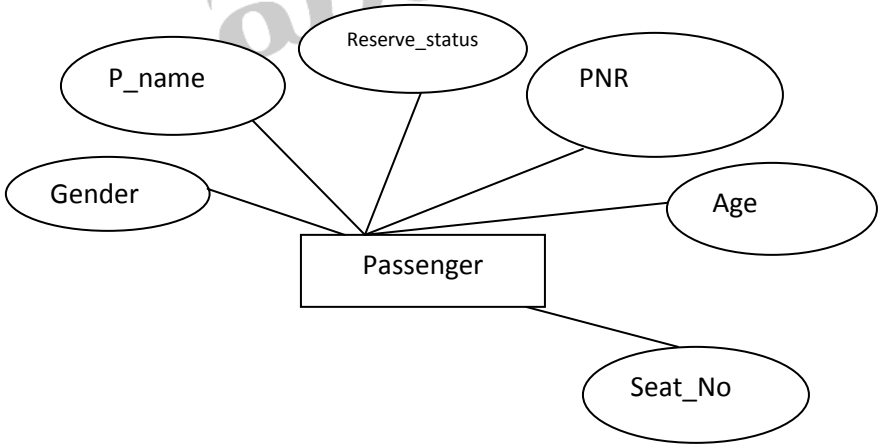
Route (Entity)



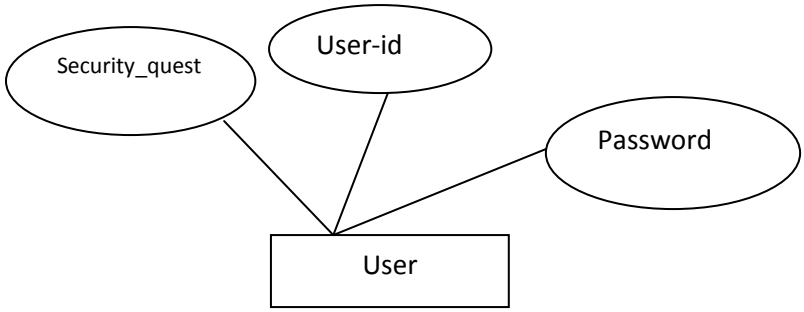
Station (Entity)



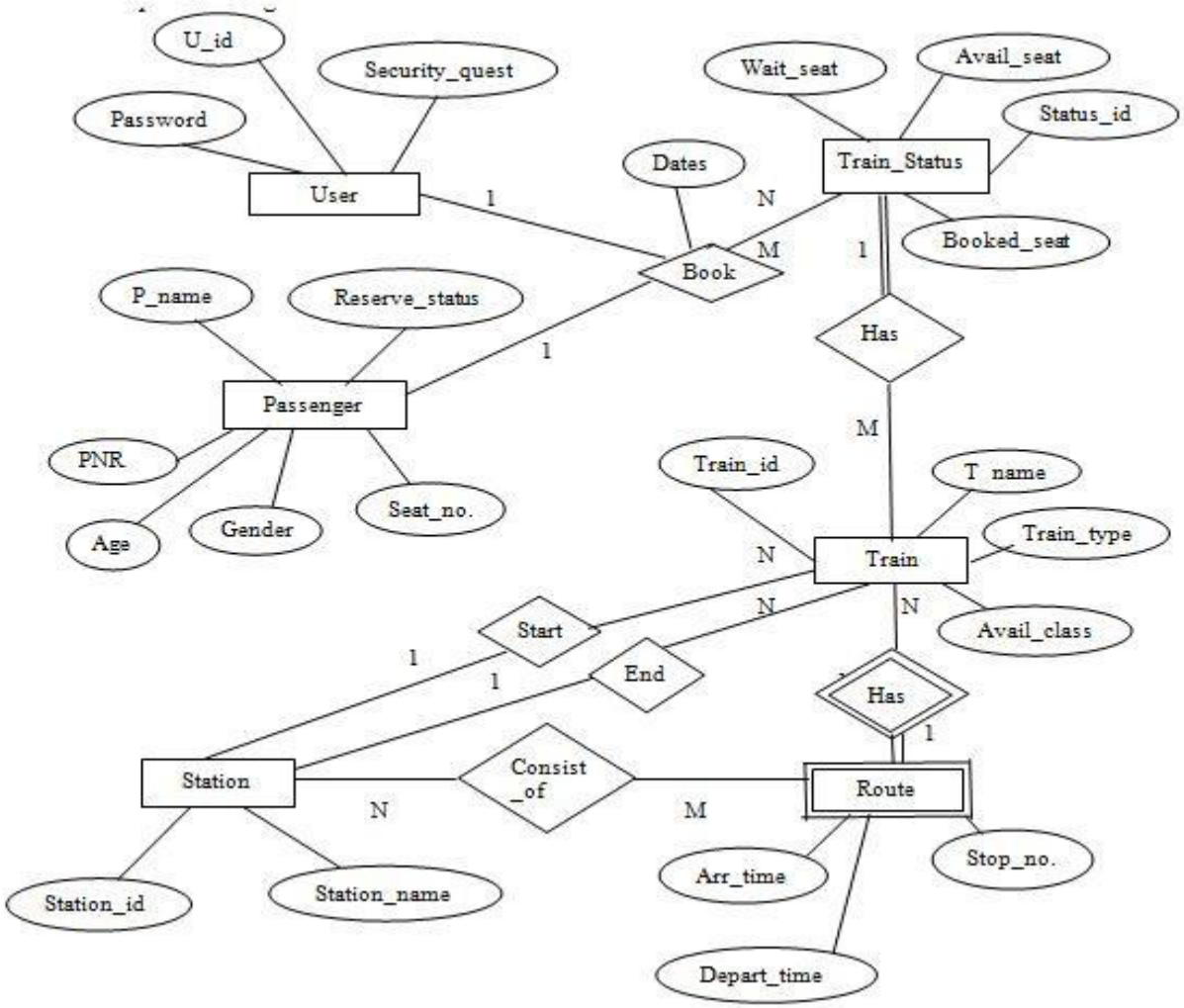
Passanger (Entity)



User (Entity)



Concept design with E-R Model:



Identifying the relationships

a. Passenger can book many tickets.

Therefore the relations are 1……..N.

b. Trains are associated with each Train_status

Therefore the relation is M……..1.

c. Trains can route from one route.

Therefore the relations are N……..1.

d. Station has a set of Route.

Therefore the relations are 1……..N.

2. Hospital Administration

Aim: To draw ER diagram for Hospital Administration.

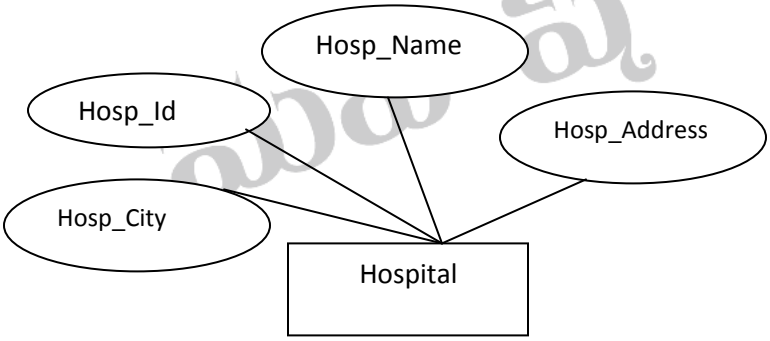
The following are the entities:

1. Hospital
2. Doctor
3. Patient
4. Medical Record

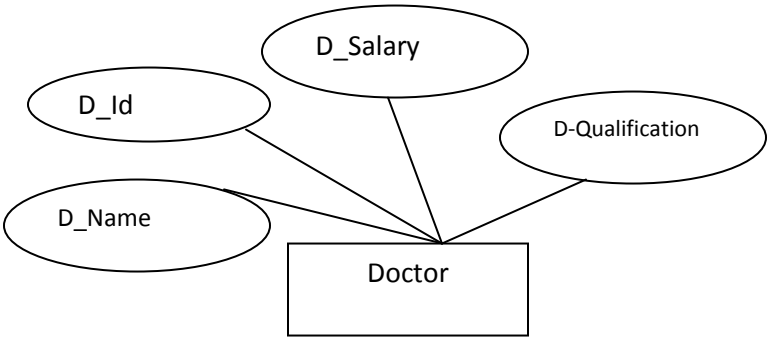
Procedure

The attributes in the entities

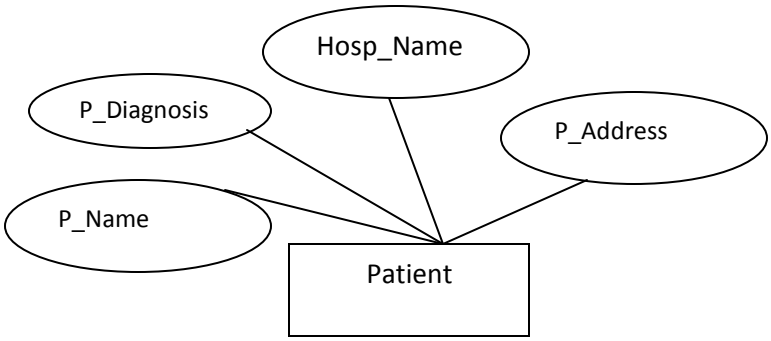
Hospital (Entities)



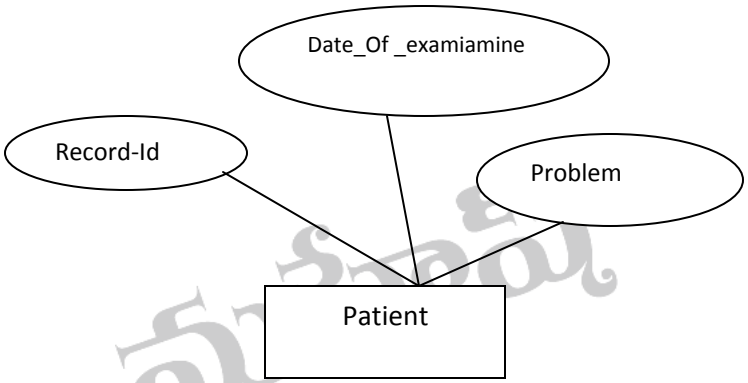
Doctor (Entities)



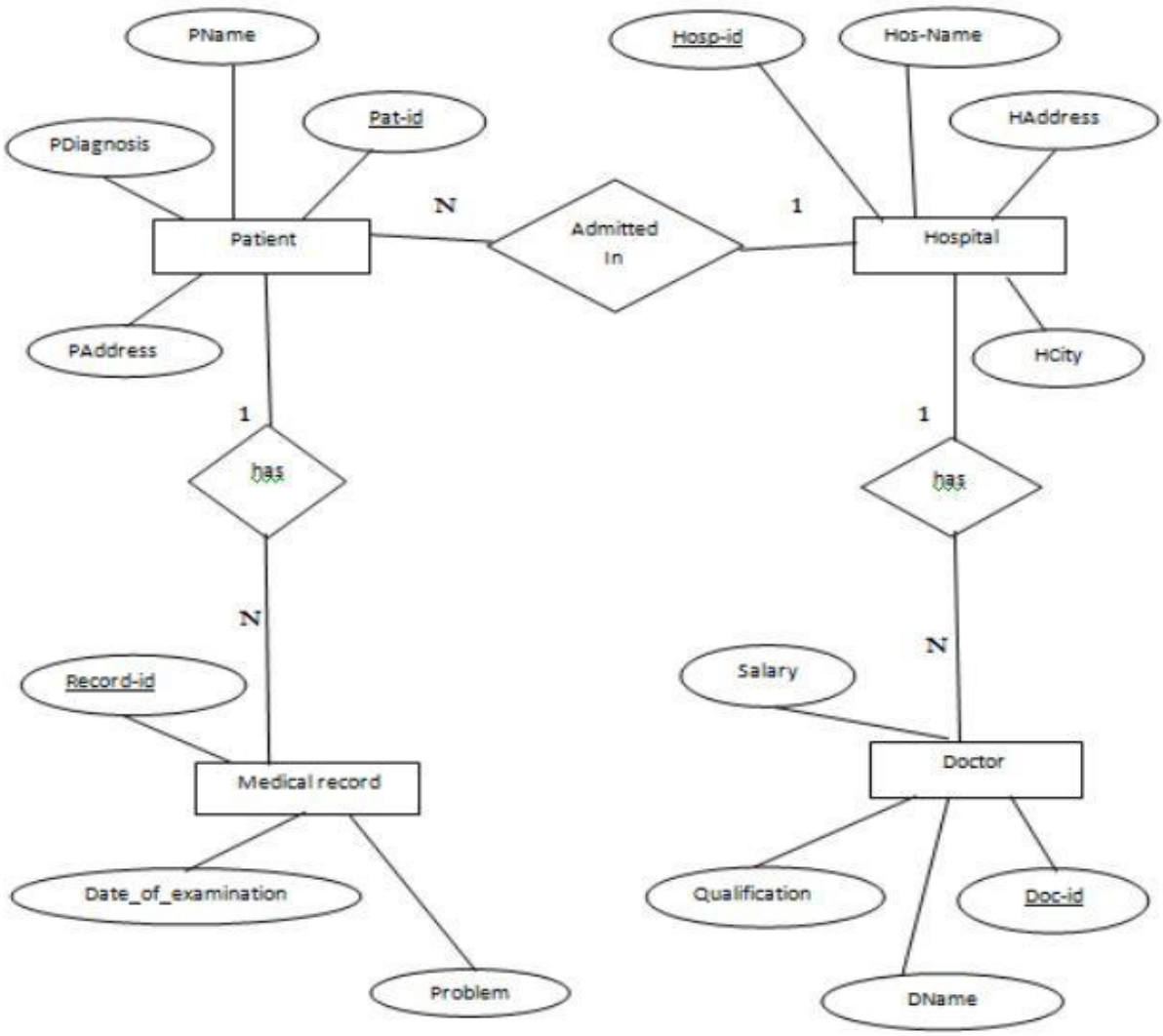
Patient (Entities)



Medical Record (Entities)



Concept design with E-R Model:



Identifying the relationships

a. Hospital has a set of patients.

Therefore the relations is 1……..N.

b. Hospital has a set of doctors.

Therefore the relations is 1……..N.

c. Doctor are associated with each patient.

Therefore the relations is N……..1.

d. Each patient has record of various test and examination conducted.

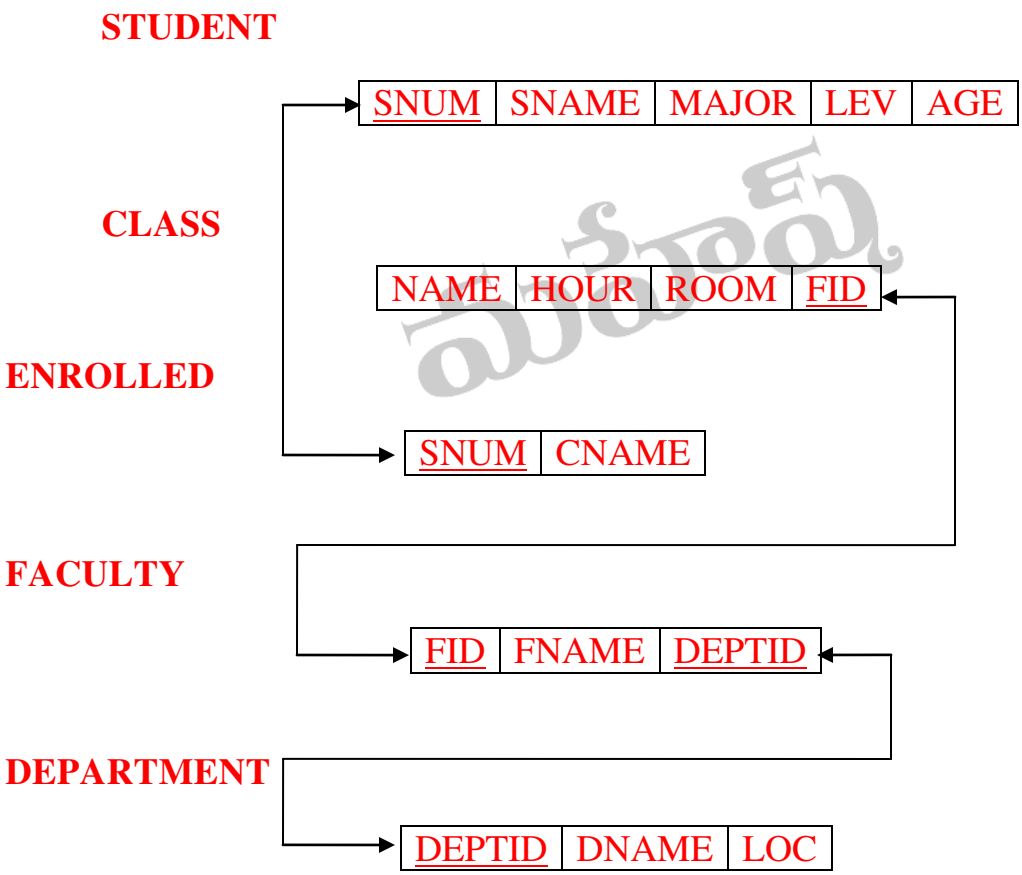
Therefore the relations is 1……..N.

3. College Database

Aim: To create a college database and establish relationship between tables.

Colleeg wishes to computerise their operations by using the following relations.
Student (snum:Integer, sname: string, major: string, level: string, age: integer)
Class (name: String, Hour:Integer, room: string, fid: integer)
Enrolled (sum: integer, cname: string)
Faculty (fid: Integer, fname: String, deptid: Integer)
Depart (deptid: Integer, dname: String, loc: integer)

Procedure:



Step1: Create a student table with the following column names and constraints

Sql>create table **student**(snum number(5),sname varchar2(15),major varchar2(15),lev varchar2(15),age number(5))

Step2: Values insertion into student table

Sql>insert into **student** values(&snum,'&sname','&major','&lev',&age)

SQL> select *from student;

SNUM	SNAME	MAJOR	LEV	AGE
24001	rajesh	computers	sr	25
24002	kumar	botany	sr	23
24003	priya	botany	jr	23
24004	ramesh	computers	sr	24
24005	nithya	english	jr	23

Step3: Create a **class** table with the following column names and constraints

Sql>create table **class**(name char(15),hour number(8),room varchar2(15),fid number(8))

Step4: Values insertion into **Class** table

Sql>insert into **class** values('&name',&hour,'&room',&fid)

SQL> select *from class;

NAME	HOUR	ROOM	FID
computers	2	1_25	200
maths	1	1_25	201
english	5	1_25	204
maths	6	1_25	205

Step5: Create a **enrolled** table with the following column names and constraints

Sql>create table **enrolled**(snum number(5), cname varchar2(15))

Step6: Values insertion into **Enrolled** table

Sql>insert into **enrolled** values(&snum,'&cname')

SQL> select *from enrolled;

SNUM	CNAME
24001	telugu
24002	computers
24003	english
24005	computers

Step7: Create a **Faculty** table with the following column names and constraints

Sql>create table **faculty**(fid number(3),fname varchar2(30),deptid number(5))

Step8: Values insertion into **Enrolled** table

Sql>insert into **faculty** values(&fid,'&fname',&deptid)

SQL> select *from faculty;

FID	FNAME	DEPTID
200	madhu	300
202	kaladhar	301
203	murali	303
205	Shruthi	300

Step9: Create a **Department** table with the following column names and constraints

Sql>create table **depart**(deptid number(5),dname varchar2(15),loc varchar2(15))

Step10: Values insertion into **Depart** table

Sql>insert into **depart** values(&deptid,'&dname','&loc')

SQL> select *from depart;

DEPTID	DNAME	LOC
300	telugu	block1
301	english	block2
303	computers	block34

4.Employee Database

Aim: An enterprise wishes to maintain a database to automate its operations. Enterprise divided into to certain departments and each department consists of employees. The following two tables describes the automation schemas.
DEPT (DEPTNO, DNAME, LOC)
EMP (EMPNO,ENAME,JOB,MGR,HIREDATE,SAL,COMM,DEPTNO)

Step1: Create department table with the following column names and constraints.
Sql>create table dept(deptno number(3) primary key,dname varchar2(15),loc varchar2(15))
Step2: values insertion into department table
Sql>insert into dept values(&deptno,'&dname','&loc')
select *from dept;

DEPTNO	DNAME	LOC
10	accounting	new york
20	research	dallas
30	sales	chicago
40	operations	boston

Step3: Create employee table with the following column names and constraints.

Sql>create table emp(eno number(4) primary key,ename varchar2(10),job varchar2(10) not null,mgr number(4),hdate date,sal number(8,2)check(sal between 15000 and 25000),comm number(4),deptno number(3),foreignkey(deptno) references dept(deptno))
Step4: Values insertion into employee table.

Sql>insert into emp values(&eno,'&ename','&job',&mgr,'&hdate',&sal,&comm,&deptno)

Sql>insert into emp (eno,ename,job,mgr,hdate,sal,deptno)
values(&eno,'&ename','&job',&mgr,'&hdate',&sal,&deptno)

ENO	ENAME	JOB	MGR	HDATE	SAL	COMM	DEPTNO
101	smith	clerk	110	17-DEC-10	8000	300	20
102	allen	salesman	110	20-FEB-11	16000	500	30
103	ward	salesman	111	09-JAN-11	12500	500	10
110	scott	manager	105	09-MAR-09	18750		10
111	king	manager	111	08-AUG-09	43750		10
105	james	manager	110	09-AUG-10	10000		20

Queries

1.DETERMINE THE EMPLOYEES, WHO LOCATED AT THE SAME PLACE.

select * from emp where deptno in(select deptno from dept where loc='&loc')

Enter value for loc: chicago

old 1: select * from emp where deptno in(select deptno from dept where loc='&loc')

new 1: select * from emp where deptno in(select deptno from dept where loc='chicago')

ENO	ENAME	JOB	MGR	HDATE	SAL	COMM	DEPTNO
102	allen	salesman	110	20-FEB-11	16000	500	30

2.DETERMINE THE EMPLOYEES, WHOSE TOTAL SALARY IS LIKE THE MINIMUM SALARY OF ANY DEPARTMENT.

select ename from emp where sal=(select min(sal) from emp)

ENAME
---
smith
james

3. UPDATE THE EMPLOYEE SALARY BY 25%, WHOSE EXPERIENCE IS GREATER THAN 1 YEAR

update emp set sal=sal+0.25*sal where months_between(sysdate,hdate)/12>1

4. DELETE THE EMPLOYEES, WHO COMPLETED 32 YEARS OF SERVICE.

delete from emp where months_between(sysdate,hdate)/12>3

5. DETERMINE THE DEPARTMENT DOES NOT CONTAIN ANY EMPLOYEES.

select *from dept where deptno in(select deptno from dept minus select distinct deptno from emp)

DEPTNO	DNAME	LOC
40	operations	boston

5. Sum and average and Biggest of three number

Aim: To find the sum and average upto given range.

```
declare
s number(3):=0;
n number(2):=&n;
k number(4,2);
begin
for j in 1 .. n
loop
s:=s+j;
end loop;
k:=s/n;
dbms_output.put_line('sum is ' ||s);
dbms_output.put_line('avg is ' ||k);
end;
/
```

Aim: To find biggest of three numbers

PROGRAM FOR FINDING BIGGEST OF THREE NUMBERS

```
declare
a number(4):=&a;
b number(4):=&b;
c number(4):=&c;
big number(4);
begin
if (a>b) then
big:=a;
else
big:=b;
end if;
if(c>big) then
big:=c;
end if;
dbms_output.put_line('biggest number ' || big);
end;
/
```

6.Cursors

Aim: TO WRITE A PL/SQL PROGRAM TO RAISE THE EMPLOYEE SALARY BY 10%, FOR DEPARTMENT NUMBER 30 PEOPLE AND ALSO MAINTAIN THE RAISED DETAILS IN THE RAISE TABLE.

Procedure

Sql>create table empm(eno number(4),ename varchar(15),job varchar2(10),hdate date,sal number(10,2),comm. Number(4),deptno number(4))

SQL> select *from emp;

ENO	ENAME	JOB	MGR	HDATE	SAL	COMM	DEPTNO
101	smith	clerk	110	17-DEC-10	8000	300	20
102	allen	salesman	110	20-FEB-11	15355.19	500	30
103	ward	salesman	111	09-JAN-11	8226.41	500	10
110	scott	manager	105	09-MAR-09	9871.7		10
111	king	manager	111	08-AUG-09	23033.96		10
105	james	manager	110	09-AUG-10	8000		20
5	uma				7000		50

……..----------

Sql>create table **remp**(teno number(4),tename varchar2(15),tjob varchar2(10),tsal number(8,2))

Program

```
declare
cursor c1 is select eno,ename,job,sal from emp where deptno=10;
teno emp.eno%type;
tename emp.ename%type;
tjob emp.job%type;
tsal emp.sal%type;
//rsal emp.sal%type;
begin
open c1;
loop
fetch c1 into teno,tename,tjob,tsal;
update emp set sal=sal+sal*0.3 where deptno=10;
insert into remp values (teno,tename,tjob,tsal);
commit;
exit when c1%notfound;
end loop;
close c1;
end;
/
```

Output:

SQL> select *from emp;

SQL> select *from remp;

TENO	TENAME	TJOB	TSAL
102	allen	salesman	36317.13
102	allen	salesman	36317.13

6. Electricity Bill

Aim: WRITE A PL/SQL PROCEDURE TO PREPARE AN ELECTRICITY BILL BY USING FOLLOWING TABLE

TABLE USED: ELECT

NAME	*NULL?*	*TYPE*
MNO	NOT NULL	NUMBER(3)
CNAME		VARCHAR2(20)
CUR_READ		NUMBER(5)
PREV_READ		NUMBER(5)
NO_UNITS		NUMBER(5)
AMOUNT		NUMBER(8,2)
SER_TAX		NUMBER(8,2)
NET_AMT		NUMBER(9,2)

Procedure

Sql>create table elect(mno number(3)primary key, cname varchar2(10),cur_read number(5),prev_read number(5),no_units number(8,2),amount number(8,2),ser_tax number(8,2),net_amt number(9,2))

Sql>insert into
elect(mno,cname,cur_read,prev_read)values(&mno,'&cname',&cur_read,&prev_read)
/
SQL> select *from elect;

MNO	CNAME	CUR_READ	PREV_READ	NO_UNITS	AMOUNT	SER_TAX	NET_AMT

101	madhu	1500	1000
105	manohar	1700	750
103	anasuya	1750	350
104	padma	3500	2000
102	ramesh	1500	500

Program

```
declare
cursor c1 is select *from elect;
rec elect%rowtype;
begin
open c1;
loop
fetch c1 into rec;
exit when c1%notfound;
rec.no_units:=rec.cur_read-rec.prev_read;
if rec.no_units>=1000 then
rec.amount:=rec.no_units*5.25;
else
rec.amount:=rec.no_units*3.25;
end if;
if rec.amount>=1500 then
rec.ser_tax:=25;
else
rec.ser_tax:=15;
end if;
rec.net_amt:=rec.amount+rec.ser_tax;
update elect set no_units = rec.no_units,
amount=rec.amount,ser_tax=rec.ser_tax,net_amt=rec.net_amt where mno=rec.mno;
end loop;
close c1;
end;
/
```

Output

```
SQL> select *from elect;
MNO  CNAME  CUR_READ  PREV_READ   NO_UNITS    AMOUNT    SER_TAX    NET_AMT
---------- ---------- ---------- ---------- ---------- ---------- ---------- ----------
 101    madhu        1500       1000        500        1625        25        1650
 105    manohar      1700       750         950        3087.5      25        3112.5
 103    anasuya      1750       350         1400       7350        25        7375
 104    padma        3500       2000        1500       7875        25        7900
 102    ramesh       1500       500         1000       5250        25        5275
```

8. Student results

AIM: WRITE A PL/SQL PROCEDURE TO EVALUATE THE GRADE OF A STUDENT WITH FOLLOWING CONDITIONS:

i. FOR PASS: ALL MARKS > 40
ii. FOR I CLASS: TOTAL%>59
iii. FOR II CLASS: TOTAL% BETWEEN >40 AND <60
iv. FOR III CLASS: TOTAL% =40

AND ALSO MAINTAIN THE DETAILS IN ABSTRACT TABLE.

TABLES USED

TABLE STD

```
SQL> DESC STD

 NAME                      NULL?    TYPE
 ------------------------------ -------- ----
 NO                      NOT NULL NUMBER
 NAME                             VARCHAR2(10)
 INTNO                            NUMBER
 CLASS                   NOT NULL VARCHAR2(10)
 M1                               NUMBER
 M2                               NUMBER
 M3                               NUMBER
 M4                               NUMBER
 M5                               NUMBER
```

TABLE ABSTRACT

```
SQL> DESC ABSTRACT
 NAME                      NULL?    TYPE
 ------------------------------ -------- ----
 STDNO                            NUMBER
 STDNAME                          VARCHAR2(10)
 CLASS                            VARCHAR2(10)
 INTNO                            NUMBER
 TOT                              NUMBER
 GRADE                            VARCHAR2(10)
 PERCENT                          NUMBER
 DAT_ENTER                        DATE
```

Procedure

```
Sql>create table student(no  number(3)primary key,name varchar2(10),intno number(3),class
varchar2(10)not null,m1 number(3),m2 number(3),m3 number(3),m4 number(3),m5 number(3))
/
```

```
Sql>insert into student values(&no,'&name',&intno,'&class',&m1,&m2,&m3,&m4,&m5)
/
SQL> select *from student;

      NO         NAME       INTNO     CLASS          M1       M2       M3       M4       M5
---------- ---------- ---------- ----- ----- ---------- ---------- ---------- ---------- ----------
       101        priya         10      Bsc          50       55       60       76       50
       102       Aastitha       11     B.com         40       67       89       90       68
       103        madhu         12      BA           66       77       88       99       78
       104        balu          13      Bsc          45       67       87       65       90
       105        anusha        14     bcom          67       78       90       87       65
Sql>create table abstract(stdno number(3),stdname varchar2(10),class varchar2(10),intno number(4),tot
number(4),grade varchar2(10),percent number(6,2))
/
SQL> desc abstract;
 Name                                      Null?    Type
 ----------------------------------------- -------- ------------
 STDNO                                              NUMBER(3)
 STDNAME                                            VARCHAR2(10)
 CLASS                                              VARCHAR2(10)
 INTNO                                              NUMBER(4)
 TOT                                                NUMBER(4)
 GRADE                                              VARCHAR2(10)
 PERCENT                                            NUMBER(6,2)
```

PL/SQL Procedure

```
declare
cursor stu_c is select *from student;
s1 student%rowtype;
total number(4);
grade1 varchar2(10);
per1 number(6,2);
begin
open stu_c;
loop
fetch stu_c into s1;
exit when stu_c%notfound;
total:=s1.m1+s1.m2+s1.m3+s1.m4+s1.m5;
per1:=total/5;
if s1.m1<40 or s1.m2<40 or s1.m3<40 or s1.m4<40 or  s1.m5<40 then
grade1:='fail';
elsif per1=40 then
grade1:='III class';
elsif per1>40 and per1<60 then
grade1:='II class';
```

```
elsif per1>59 then
grade1:='I class';
end if;
insert into abstract  values(s1.no,s1.name,s1.class,s1.intno,total,grade1,per1);
end loop;
close stu_c;
end;
/
```

Output:

```
SQL> select *from abstract;
```

STDNO	STDNAME	CLASS	INTNO	TOT	GRADE	PERCENT
101	priya	Bsc	10	291	II class	58.2
102	Aasritha	B.com	11	354	I class	70.8
103	madu	BA	12	408	I class	81.6
104	balu	Bsc	13	354	I class	70.8
105	anusha	bcom	14	387	I class	77.4